

Ultra-low-power Physical Activity Classifier for Wearables: From Generic MCUs to ASICs

Enric M. Calvo, Philippe Renevey, Mathieu Lemay, Andrea Bonetti,
Marc Pons Solé, Régis Cattenoz, Stéphane Emery, and Ricard Delgado-Gonzalo

Abstract—In the era of Internet of Things (IoT), an increasing amount of sensors is being integrated into intelligent wearable devices. These sensors have the potential to produce a large quantity of physiological data streams to be analyzed in order to produce meaningful and actionable information. An important part of this processing is usually located in the device itself and takes the form of embedded algorithms which are executed into the onboard microcontroller (MCU). As data processing algorithms have become more complex due to, in part, the disruption of machine learning, they are taking an increasing part of MCU time becoming one of the main driving factors in the energy budget of the overall embedded system. We propose to integrate such algorithms into dedicated low-power circuits making the power consumption of the processing part negligible to the overall system. We provide the results of several implementations of a pre-trained physical activity classifier used in smartwatches and wristbands. The algorithm combines signal processing for feature extraction and machine learning in the form of decision trees for physical activity classification. We show how an in-silicon implementation decreases up to 0.1 μW the power consumption compared to 73 μW on a general-purpose ARM's Cortex-M0 MCU.

I. INTRODUCTION

Wearable devices provide the user with a low-cost solution to continuously monitor physiological parameters in an inconspicuous way. This domain has experienced a rapid growth in the last decade, in part, as a consequence of the Quantified Self movement who pledges to track personal data and physiological signals under the premise to develop healthy behaviors in the user [1]. Wearable devices have achieved great success both for personal healthcare management (*e.g.*, wristbands [2], smart-vests [3]) and for support to clinical treatments [4]. In this context, autonomy (*i.e.*, battery life) is a key factor in order for the device to be able to monitor the user for long periods of time [5].

In the context of human kinetics, wearable devices mostly focus on detecting, classifying, and profiling the kinetic information of the wearer gathered with inertial sensors [6]. In this space, accelerometers have taken the lead due to their good balance between kinetic information acquired, power consumption, cost, and miniaturization. Moreover, the most prevalent everyday activities (resting, walking, and running) can successfully be classified with high precision and recall [7], [8].

From a system's perspective, a wearable device that continuously monitors physiological data can be roughly

simplified to two interlocked blocks: a block containing a specific set of sensors capable of capturing raw data and a block containing the algorithm, or set of algorithms, capable of processing such data in order to provide actionable data. Traditionally the processing of raw sensor signals has taken place in the microcontroller with algorithms designed to the specific application. Accelerometers have improved up to two orders of magnitude their power consumption in the past decade [9], rendering the algorithmic processing in the MCU the most power-hungry element in the chain. By integrating the algorithms directly into the sensor through a dedicated Application-Specific Integrated Circuit (ASIC) instead of deferring the computation to the MCU, physical activity tracking wearables would benefit from extended battery life and would enable an uninterrupted data analysis even when energy is scarce.

In the present study, we present the results of adapting a physical activity tracker algorithm into several increasingly low-power architectures. The algorithm relies exclusively on an accelerometer as data input, operates in a sample-by-sample basis, and combines signal processing for feature extraction and machine learning (pre-trained decision trees) for the inference of the physical activity. We evaluate the tradeoffs between power consumption and flexibility of the algorithm implementation into two off-the-shelf microcontrollers (one containing a Floating Point Unit and the other without, and two dedicated ASIC implementations (a low power processor and an optimized hardware accelerator).

In the following section, we describe the architecture of the algorithm as well as the data in which it has been trained. Then, we proceed to detail the evaluation metrics and the different hardware architectures that have been tested. And finally, we compare the results.

II. MATERIALS

In this section, we provide an overview of the algorithm that we used in our analysis, and then, we provide details on the data that was used to train some of its blocks.

A. Activity algorithm overview

The physical activity tracking algorithm takes as input the raw 3-axis accelerometer signals at 25 Hz and outputs the most likely activity among the following classes: Rest, Walk, Run, Bike, or Other. It can be grouped into 6 blocks, as shown in Figure 1. The first 5 blocks are based on signal processing and dedicated towards extracting signal features that discriminate the different physical activities. The last

The authors are with the Centre Suisse d'Electronique et de Microtechnique (CSEM), Jacquet-Droz 1, Neuchâtel, Switzerland (e-mail: ricard.delgado@csem.ch).

block contains a balanced binary decision tree based on the aforementioned features. More precisely the blocks are:

- *Signal Conditioning*: The raw accelerometer signal is low-passed filter in order to remove high-frequency noise.
- *Coordinates Conversion*: The three coordinates are transformed and combined in vector form in order to be independent from sensor location and orientation.
- *Multiple Filtering Stages*: Several non-linear filters are applied to reduce outliers and to create time-consistency.
- *Subband Splitting*: The main feature signal is split into 5 frequency sub-bands.
- *Tracking Filters*: On each sub-band, the main frequency is identified and tracked.
- *Classifier*: A binary classification tree is used to estimate, for each sample, the likelihood of such sample belonging to one of the physical activity classes. The classifier uses the features extracted in the previous block and was trained offline with the data detailed in Section II-B. Further details on the training of the algorithm can be found in [10].

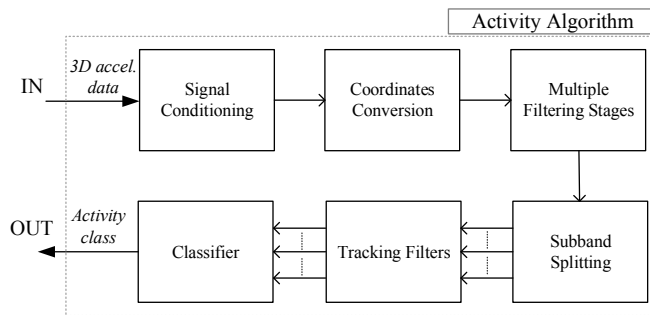


Fig. 1: Block diagram of the physical activity algorithm.

In Figure 2, we show the response of the described algorithm to a change of physical activity. In particular, it illustrates the transition between *Walk* and *Run*. A visual inspection of the acceleration signals shows that the transition occurs approximately at $t = 27$ s. At that point in time, the likelihood of the *Run* class starts to raise smoothly and the likelihood of the *Walk* class starts to drop smoothly. The algorithm outputs the class with the highest likelihood as long as the likelihood is higher than 50%, otherwise, the algorithm outputs *Other*. Note that the latency in the estimation of the activity with this example amounted to approximately 3 seconds. This lag is determined by the coefficient parameters of the 3rd block of the algorithm.

B. Data

In order to train the algorithm, a smart wrist-band integrating a three-axial accelerometer from PulseOn was used¹. Inertial signals were collected with a sampling frequency of 25 Hz, 12 bit resolution, and $\pm 8g$ range.

The acceleration forces from the wrist-located sensors were recorded on 140 individuals (76 male, 64 female) in 18

¹<https://pulseon.com/>

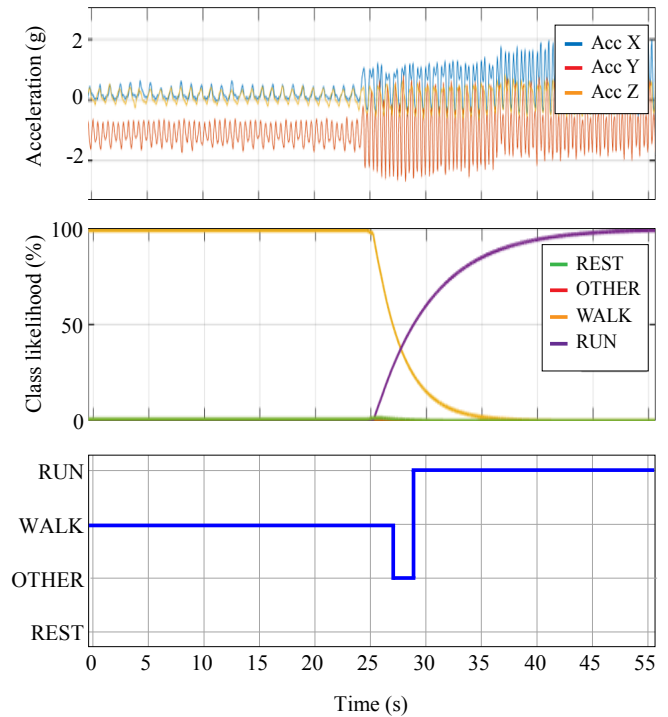


Fig. 2: Output of the physical activity algorithm to a change of activity. (top) Raw accelerometer signals, (middle) Smoothed probabilities of each class, (bottom) Output of the classifier

recording campaigns. The data collection was conducted between 2014 and 2017 in Tampere (Finland), Espoo (Finland), and Neuchâtel (Switzerland)² and included in-lab protocols and real-life activities. A total number of 418 recordings spanning more than 440 hours of raw data was gathered.

III. METHODS

In this section, we discuss the algorithm's implementation at different levels of abstraction: out-of-the-shelf System-on-Chip (SoC) MCUs, on an application-specific instruction set processor, and on an ASIC.

A. Out-of-the-shelf microcontrollers

In [10], the algorithm was implemented in C using fixed-point arithmetic as well as floating-point arithmetic. Each implementation targeted a different SoC. The fixed-point implementation was conceived for systems without Floating Point Units (FPU) on its core. For that purpose the version in fixed-point arithmetic implemented manually all 32-bit divisions as bitwise shifts. In particular, the algorithm was benchmarked on a Nordic Semiconductor's nRF51832 and nRF52832 SoC using an ARM's Cortex-M0 and Cortex-M4f core respectively. The compilation was carried over with arm-gcc with -O3 optimization.

²The experimental procedures described in this paper complied with the principles of Helsinki Declaration of 1975, as revised in 2000. All subjects gave informed consent to participate and they had a right to withdraw from the study at any time. Their information was anonymized prior the analysis.

B. Application-specific instruction set processor

The second approach consists in running the existing fixed-point implementation of the algorithm into an application specific resource-constrained microcontroller designed with ultra-low power consumption in mind. We chose to use our icyflex-V processor [11] that is RISC-V³ compatible and that is optimized for low power. The icyflex-V is a classical 4-stage pipeline implementing the RV32IMC instruction set architecture (ISA). Data forwarding is implemented to avoid pipeline stalls on both arithmetic logic unit (ALU) outputs and load-store (LSU) outputs caused by read-after-write dependencies. The main target of the core is to be embedded efficiently with a small silicon footprint in ultra-low power system-on-chips. Thus, the main criteria for icyflex-V design are an excellent code density (thanks to compressed instruction) and a limited gate-count. Several actions were further taken to reduce the gate-count: the M extension of the ISA is implemented with fast and low-power hardware multiplier. Similarly, floating-point operation is left as software emulation. In order to achieve state-of-the-art performance though, the 4-stage pipeline was preferred over a 2-stage pipeline that would assuredly be more compact but which would have degraded performances drastically. Similarly, the instruction prefetch buffer implements a simple yet effective branch prediction mechanism that improves the core efficiency at a reasonable gate-count cost. Note that the operations needed by the algorithm are well supported by such processor hardware.

C. Model-based activity tracking algorithm accelerator

In this approach, the goal is to design a dedicated hardware accelerator for the algorithm. For that, we started from a fixed-point C implementation of the algorithm and generated the equivalent Hardware Description Language (HDL) code using SimulinkTM from MathWorks. The testing capabilities of SimulinkTM enabled us to perform a bit-true simulation of the final implementation. This method results in a faster hardware design any time there is a new feature to implement. Critical blocks from the generated HDL were then optimized (*e.g.*, CORDIC part). This model-based activity tracking algorithm accelerator option is the most promising for low power and small silicon area footprint because it results in a fine-tuned dedicated hardware accelerator implementing the algorithm with the minimum hardware, but it is the least flexible of all (as opposed to the processor approach in Section III-B, where algorithm software can be updated and be executed on the same hardware).

D. Implementation assessment using sub-threshold design

An extra refinement can be applied to the implementations presented in Sections III-C and III-B in order to further reduce power consumption. We used the ultra-low-power technology known as sub-threshold design [12]. This technology uses the transistors in the sub-threshold region by reducing the supply voltage (*e.g.*, from 1V to 0.5V).

This supply voltage reduction results in the main advantage of sub-threshold design for ultra-low power applications, that is that dynamic consumption quadratically decreases with the supply voltage (*e.g.*, 4x reduction can be achieved when going from 1V to 0.5V). Consequently, battery life can be extended and battery size can be reduced, which is key for wearable devices. Moreover, at this low voltage levels, designs are very well suited to run up to the MHz regime, which is a perfect match for the activity tracking algorithm. However, circuits become more sensitive to process, voltage and temperature variations (*e.g.*, 2 orders of magnitude difference in terms of frequency when comparing best versus worst case conditions). For that last reason, for this work we have chosen specifically tailored standard cells and memories for sub-threshold design that make use of body bias to compensate for these process, voltage and temperature variations. Previous systems using this technology have been shown to consume as low as 2.5 $\mu\text{W}/\text{MHz}$ [13].

IV. RESULTS

We show in Table I the power necessary to run each implementation of the algorithm on their corresponding hardware. In order to make the measurements comparable, we defined a standard testing dataset that recreates the daily use of an active user. This dataset contained 15 hours of sleep/rest, 4 hours of walking, 1 hour of running, and 4 hours of undefined activities. In order to provide a benchmark, we also provide the power necessary to run the 3D accelerometer ADXL363 from Analog Devices. Its power consumption is 4 μW which puts him in the range of the new generation ultra-low-power accelerometers.

As shown in Table I, the power consumption obtained with the general-purpose MCUs (Cortex-M0 and Cortex-M4f) dominates the column of the power consumption. Note that in the worst case (Cortex-M0), the energy spent by the MCU is 22.6 times greater than the accelerometer. A significant diminution in power consumption can be observed when using the icyflex-V technology and the dedicated hardware. Note, however, that this diminution is compensated by a reduction in flexibility. This means that the algorithm can be modified at any time in the generic microcontroller whereas the accelerator has the algorithm burned in-silicon.

The results improve even further when we use sub-threshold technology at 0.5V (two last rows in Table I). The accelerator consumption stays well below the consumption of the algorithm in a standard microcontroller (by around three orders of magnitude). The dynamic part becomes negligible due to low toggling activity and low frequency of operation. This makes very low leakage and low voltage technologies suitable for this application. The estimated power also stays well below the 4 μW consumption for a state of the art accelerometer, rendering the intelligence attached to the sensor essentially free in power terms.

V. CONCLUSIONS

In this study, we have presented a variety of implementations of a pre-trained physical activity classifier used in

³<https://riscv.org/>

TABLE I: Power consumption estimation depending on implementation.

Hardware	Techno. voltage supply	Power used by the algorithm	Algorithm to sensor power ratio	Flexibility
Cortex-M0	1.8 V	90.7 μ W	22.6	High
Cortex-M4f	1.8 V	27.5 μ W	6.9	High
icyflex-V	1.0 V	16.0 μ W	4	Medium
Accelerator	1.0 V	1.0 μ W	0.25	Low
icyflex-V	0.5 V	0.8 μ W	0.2	Medium
Accelerator	0.5 V	0.1 μ W	0.025	Low

smartwatches and wristbands. The implementations range from general-purpose MCUs to dedicated ASICs. We showed how an in-silicon implementation decreases up to 0.1 μ W the power consumption compared to 73 μ W on a general-purpose ARM's Cortex-M0 MCU. These figures correspond to 22.6 times and 0.025 times the power consumption of a state of the art accelerometer from Analog Devices.

Such decrease in power consumption to run the physical activity tracking algorithm is the next step in the new generation of battery-powered wearable sensors. We believe that this new architecture opens a new array of possibilities for health-related applications, wearables, continuous data monitoring, and energy-harvesting-based devices, where the least amount of battery or energy suffices to keep recording metrics, statistics and data without gaps. This will not only free the user from tedious battery-charging each of its IoT devices, but will ensure continuous data monitoring without current hassles.

REFERENCES

[1] J. Dunn, R. Runge, and M. Snyder, "Wearables and the medical revolution," *Personalized Medicine*, vol. 15, no. 5, pp. 429–448, 2018.

[2] E. C. Nelson, T. Verhagen, and M. L. Noordzij, "Health empowerment through activity trackers: An empirical smart wristband study," *Computers in Human Behavior*, vol. 62, pp. 364–374, 2016.

[3] S. Schneegass and O. Amft, *Smart textiles*. Springer, 2017.

[4] C. Mombers, K. Legako, and A. Gilchrist, "Identifying medical wearables and sensor technologies that deliver data on clinical endpoints," *British Journal of Clinical Pharmacology*, vol. 81, no. 2, p. 196, 2016.

[5] Z. Chen, W. Hu, J. Wang, S. Zhao, B. Amos, G. Wu, K. Ha, K. Elgazar, P. Pillai, R. Klatzky, D. Siewiorek, and M. Satyanarayanan, "An empirical study of latency in an emerging class of edge computing applications for wearable cognitive assistance," in *Proceedings of the 2nd ACM/IEEE Symposium on Edge Computing (SEC'17)*, 2017, pp. 1–14.

[6] R. Delgado-Gonzalo, P. Renevey, A. Lemkaddem, M. Lemay, J. Solà, I. Korhonen, and M. Bertschi, *Seamless healthcare monitoring*, 1st ed. Springer, Cham., 2018, ch. Physical Activity, pp. 413–455.

[7] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 119–128, 2006.

[8] R. Delgado-Gonzalo, P. Celka, P. Renevey, S. Dasen, J. Solà, M. Bertschi, and M. Lemay, "Physical activity profiling: Activity-specific step counting and energy expenditure models using 3D wrist acceleration," in *Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'15)*, Milan, Italy, Aug. 2015, pp. 8091–8094.

[9] H. Weinberg and N. Gadish, "Minimizing power consumption of iMEMS® accelerometers," *Analog Devices*, 2002.

[10] R. Delgado-Gonzalo, P. Renevey, A. Tarniceriu, J. Parak, and M. Bertschi, "Learning a physical activity classifier for a low-power embedded wrist-located device," in *Proceedings of the IEEE EMBS International Conference on Biomedical & Health Informatics (BHI'18)*, 2018, pp. 54–57.

[11] J.-L. Nagel, C. Arm, R. Cattenoz, H.-R. Graf, and V. Moser, "icyflex-V: A new ultra-low power processor based on RISC-V architecture," CSEM, Tech. Rep., 2019.

[12] M. Pons, J. Nagel, D. Séverac, M. Morgan, D. Sigg, P.-F. Rüedi, and C. Piguët, "Ultra low-power standard cell design using planar bulk CMOS in subthreshold operation," in *Proceedings of the 23rd International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS'13)*, Karlsruhe, Germany, Sep. 2013, pp. 9–15.

[13] M. Pons, C. T. Müller, D. Ruffieux, J.-L. Nagel, S. Emery, A. Burg, S. Tanahashi, Y. Tanaka, and A. Takeuchi, "A 0.5 V 2.5 μ W/MHz microcontroller with analog-assisted adaptive body bias PVT compensation with 3.13 nW/kB SRAM retention in 55nm deeply-depleted channel CMOS," in *Proceedings of the 2019 IEEE Custom Integrated Circuits Conference (CICC'19)*, Austin, TX, USA, Apr. 2019, pp. 1–4.